Hochschule München FK 07 - Informatik Prof. Dr. Chr. Vogt
2nd Lab Assignment for the Course "Operating Systems 1"

Topic: Watching Priority Adjustments in Windows.

Resources:

- The program **CPUSTRES.EXE** allows to run up to 4 worker threads. You can specify the *base priority* of the individual threads, as well as an *"Activity Level"*. At Activity "Low", the thread is sleeping about 75% of the time, at "Maximum" it executes an infinite loop.
- The program **IO_intensive.exe** continuously writes to the specified file on disk.
- The program **Scheduler_Test.exe** creates two threads which increment and output an integer in an infinite loop (thread 1 also counts to 10,000,000 in between outputs).
- For monitoring counters, use **PERFMON4.EXE**, which is the performance monitor from Windows NT 4.0. It allows to set the update interval to values smaller than 1 second (in the exercises, use a value of 0.01)

Exercise 1

You can view the queues of ready threads with the kernel debugger's !ready command. Use the "normal" kernel debugger kd (started with `kd –kl` to debug the live system) because livekd enters a loop when using this command.
Use this command in different situations, including the case where you forced the existence of several ready threads with the help of CPUSTRES.EXE.

Exercises 2 and 3 must be solved on a single processor system, e.g. the virtual machine on the lab computers.

Exercise 2

For this exercise, use the Process Explorer to observe running threads (in particular: current priority, CPU usage [user and kernel mode], and context switch delta), and interrupt activity (number of interrupts, and CPU time used for interrupt handling). Do not take the absolute values you observe on the virtual machine too serious. Those would be quite different on a physical machine. However, in principle, the behavior is of course the same.

a) *Foreground vs. background:*
Run CPUSTRES.EXE with one active thread. How many threads does the process have? Watch the current priority of the primary thread both when the CPUSTRES window is in the foreground, and when it is in the background. Focus on the main difference, and don't try to understand every detail of the behavior. Describe and explain your observations.

b) *Running several CPU-bound threads:*
Observe CPU usage, context switch delta, and state of the worker threads of CPUSTRES.EXE with varying numbers of worker threads, all at activity level "Maximum". In particular, answer the following questions:
- How and why do the values for context switch delta change with the number of active threads?
- What can you conclude from the values observed?
- Why do you never see any of the threads in the state "running"?

c) *Running an I/O-intensive thread:*
End CPUSTRES.EXE, and run IO_intensive.exe. Describe your observations (priority, context switch delta, CPU time in user mode, and in kernel mode resp.) Also look at the interrupt activity on the system. Describe and explain your observations.

d) *Running both a CPU-bound thread, and an I/O-intensive thread:*
Run CPUSTRES.EXE (activity maximum), and IO_intensive.exe simultaneously. How and why does running IO_intensive.exe change the behavior of the CPUSTRES.EXE threads? Does the behavior of IO_intensive.exe also change (significantly)?

e) *Running both a CPU-bound thread, and an I/O-intensive thread, without dynamic priority adjustment:*
Doing so actually worked on an earlier version of Fedora Linux as the host for the Windows virtual machine (the reason is somewhat complicated, and not in the scope of this class). The behavior of the two programs then is very different. Describe the behavior that is to be expected in this case, and evaluate the effect of the priority adjustment done by the Windows scheduler in this particular situation (which, of course, is a common one).

Exercise 3

Run Scheduler_Test.exe. Observe and describe the output. What are the base priorities of the threads? Observe the current priorities using PERFMON4.EXE. Describe and explain your observations.

Turn in a printed version of your solutions by the due date to be announced in class. Do not only describe the results, but give proof of your practical work, e.g. by including meaningful screenshots.